

KBS-203: Kubernetes Advanced Administration (Helm + Security + Istio)

Course Length: 1 day Helm, 1 day Cloud-native security + 1 day Istio, 3 days altogether

Course Description:

Kubernetes is the de-facto system for container orchestration, e.g. automating the deployment, scaling and management of microservices-based, containerized applications.

This course builds on the of Linux container and basic Kubernetes admin. skills participants gained on our Docker and Kubernetes trainings or in practice and enhances their knowledge with more advanced Kubernetes features and extensions such as the Helm package manager, major cloud-native security topics and tools to handle security related tasks as well as Istio, the most popular service mesh solution.

Structure: 50% theory 50% hands on lab exercises

Target audience: System administrators, developers and devops with Kubernetes admin. skills who want to learn about advanced Kubernetes extensions like the Helm package manager, cloud native security and service meshes such as Istio.

Prerequisites: Linux container (e.g. Docker) and Kubernetes admin. skills, for instance by participating on our Docker and Kubernetes administration courses.

Detailed Course Outline

PART I: Helm Package Manager

Module 1: Introduction to Helm

- What is Helm?
- Helm concepts
- Helm v2 (legacy) components
- Helm v3 components
- Installing Helm
- Helm Lab: Installing Helm

Module 2: Using Helm

- Generic options and help
- Working with repositories
- Finding charts
- Installing a release
- List releases
- Upgrade/rollback releases
- Uninstalling releases
- Helm Lab: Using Helm

Module 3: Helm charts

- Introduction to charts
- The structure The Chart.yaml File
- The components of a Chart
- Chart dependencies
- Chart dependencies (cont.)

- Managing Charts with helm
- Helm Lab: Working with charts

Module 4: Chart Templates

- Writing Templates Templates and Values
- Dependencies and values Dependencies and values
- Chart lifecycle hooks Functions and pipelines
- Flow control
- Variables
- Named templates
- Helm Lab: Writing templates

Module 5: Helm plugins

- Building plugins
- Helm Lab: Helm plugins

PART II: Cloud-native Security

Module 7: Managing users in Kubernetes (Authentication and permissions)

- Establishing the identity
- Using OIDC tokens with Kubernetes
- Creating our own OIDC provider with Dex
- Authorization in Kubernetes
- Role based access control (RBAC)
- Lab: Authenticate LDAP users into Kubernetes

Module 8: Secure image registries

- Security requirements for image registries
- Building secure registries using Harbor
- Harbor Operations
- Lab: Using Harbor as an image registry

Module 9: Penetration testing the Kubernetes API

- What to test?
- Tools for penetration testing
- Lab: Penetration test the Kubernetes cluster

Module 10: Monitoring application behaviour

- Mechanisms for controlling application behavior
- Tools for observing application behavior
- Observing application behavior using Falco
- Lab: Monitor application behavior in Kubernetes

Module 11: Container isolation mechanisms

- Pod security policies
- Configuring pod security policies
- Lab: Enforce security rules using Pod Security Policies

PART III: Service Mesh Essentials and Implementation with Istio

Module 12: Introduction to Istio

- Service mesh
- What is Istio?
- Istio features
- Platform support
- Istio architecture
- Istio architecture
- Istio architecture – Data plane
- Istio Architecture - Control plane
- Istio's design goals

Module 13: Installing Istio in Kubernetes

- Installing Istio on Kubernetes
- Injecting the sidecar container
- Lab:Installing Istio on Kubernetes

Module 14: Istio Traffic Management

- Traffic management
- Traffic routing concepts
- Request routing
- Discovery and load balancing
- Handling failures
- Rule configuration
- Virtual Services
- Virtual Services - examples
- Virtual Services – Timeouts and retries
- Virtual Services – Injecting faults
- Destination rules
- Destination rules – Circuit breakers
- Service entries
- Service entries - example
- Gateways
- Gateways - example
- Sidecars
- Lab:Traffic management

Module 15: Security

- Security
- Security - architecture

- Security – Identity
- Service Authentication
- Authentication policies - scope
- Authentication policies (cont)
- Authorization
- Service roles
- Service role binding
- Lab:Security

Module 16: Policies and telemetry

- Policies and telemetry
- Reliability and latency
- Attributes
- Configuration model
- Handlers
- Instances
- Rules
- Policies and telemetry